DebugAnalyser User Guide

**V 2.1**

**2017/10/26**

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Revision | Author | Reviewer |
| **2015/05/04** | Draft v1.0 | Draft |  |  |
| **2017/03/02** | Draft v1.5 | Add more Feature | Max |  |
| **2017/05/22** | Draft v1.7 | Detail Amendment | Calvin |  |
| **2017/10/26** | Draft v2.1 | Revision | Cordelia |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[Revision History 2](#_Toc497133398)

[List of figures 4](#_Toc497133399)

[List of Tables 5](#_Toc497133400)

[1 Introduction 6](#_Toc497133401)

[2 Overview of DebugAnalyser Tool 6](#_Toc497133402)

[2.1 Presentation of UI 7](#_Toc497133403)

[2.2 Preparing 7](#_Toc497133404)

[3 Configurations of DebugAnalyser Tool 7](#_Toc497133405)

[3.1 Configuration of Main Dialog 7](#_Toc497133406)

[3.2 Configuration of Detail Settings Dialog 9](#_Toc497133407)

[3.3 Start and stop capturing and decoding 10](#_Toc497133408)

[4 Log Operation and Display on UI 11](#_Toc497133409)

[4.1 Log Operation on UI 11](#_Toc497133410)

[4.2 Log Display on UI 12](#_Toc497133411)

[5 How to use Ellisys view package of BTSnoop 14](#_Toc497133412)

[5.1 Open DebugAnalyser Tool 14](#_Toc497133413)

[5.2 The Process of Using Ellisys 14](#_Toc497133414)

[5.3 Notes for Using Ellisys to view BTSnoop 17](#_Toc497133415)

List of figures

[Figure 2‑1 Overview of DebugAnalyser 5](#_Toc497122250)

[Figure 2‑2 Output Files 6](#_Toc497122251)

[Figure 3‑1 Log Decoding Type 6](#_Toc497122252)

[Figure 3‑2 Decoder Output Settings 7](#_Toc497122253)

[Figure 3‑3 Log Settings 7](#_Toc497122254)

[Figure 3‑4 Detail Settings 8](#_Toc497122255)

[Figure 3‑5 Select Bin file 9](#_Toc497122256)

[Figure 4‑1 Copy Logs 10](#_Toc497122257)

[Figure 4‑2 Show Log Details 11](#_Toc497122258)

[Figure 4‑3 Search and Filter Logs 11](#_Toc497122259)

[Figure 4‑4 Show ARM and DSP Logs 12](#_Toc497122260)

[Figure 4‑5 Show ARM Logs Only 12](#_Toc497122261)

[Figure 4‑6 Show DSP Logs Only 12](#_Toc497122262)

[Figure 5‑1 SWDTool 13](#_Toc497122263)

[Figure 5‑2 DataFile directory 14](#_Toc497122264)

[Figure 5‑3 Open Ellisys 14](#_Toc497122265)

[Figure 5‑4 Ellisys Configure 15](#_Toc497122266)

[Figure 5‑5 Start Capture 15](#_Toc497122267)

[Figure 5‑6 Open HCI Overview（Injection） 16](#_Toc497122268)

[Figure 5‑7 HCI Overview（Injection） 16](#_Toc497122269)

List of Tables

[Table 1‑1 Supported ICs 5](#_Toc497123544)

[Table 3‑1 Decoder Input selections 7](#_Toc497123545)

[Table 3‑2 Decoder Output Selections 8](#_Toc497123546)

[Table 3‑3 Serial Port Settings 9](#_Toc497123547)

[Table 3‑4 Detail Settings 10](#_Toc497123548)

# Introduction

DebugAnalyser is a tool used for capturing and decoding logs of Realtek Bluetooth SOC chip. The following ICs are supported. This document provides detailed guidance on how to set and use it.

|  |
| --- |
| Supported ICs |
| RTL8762A (Bee1)  RTL8762C (Bee2)  RTL8763B (BBPro) |

Table 1‑1 Supported ICs

The document comprises the following chapters:

**Chapter 2** provides overview of DebugAnalyser.

**Chapter 3** describes how to configure the tool and decode logs.

**Chapter 4** describes how to operate decoded output (logs) on UI.

**Chapter 5** describes how to view BT HCI log with Ellisys in real time.

# Overview of DebugAnalyser Tool

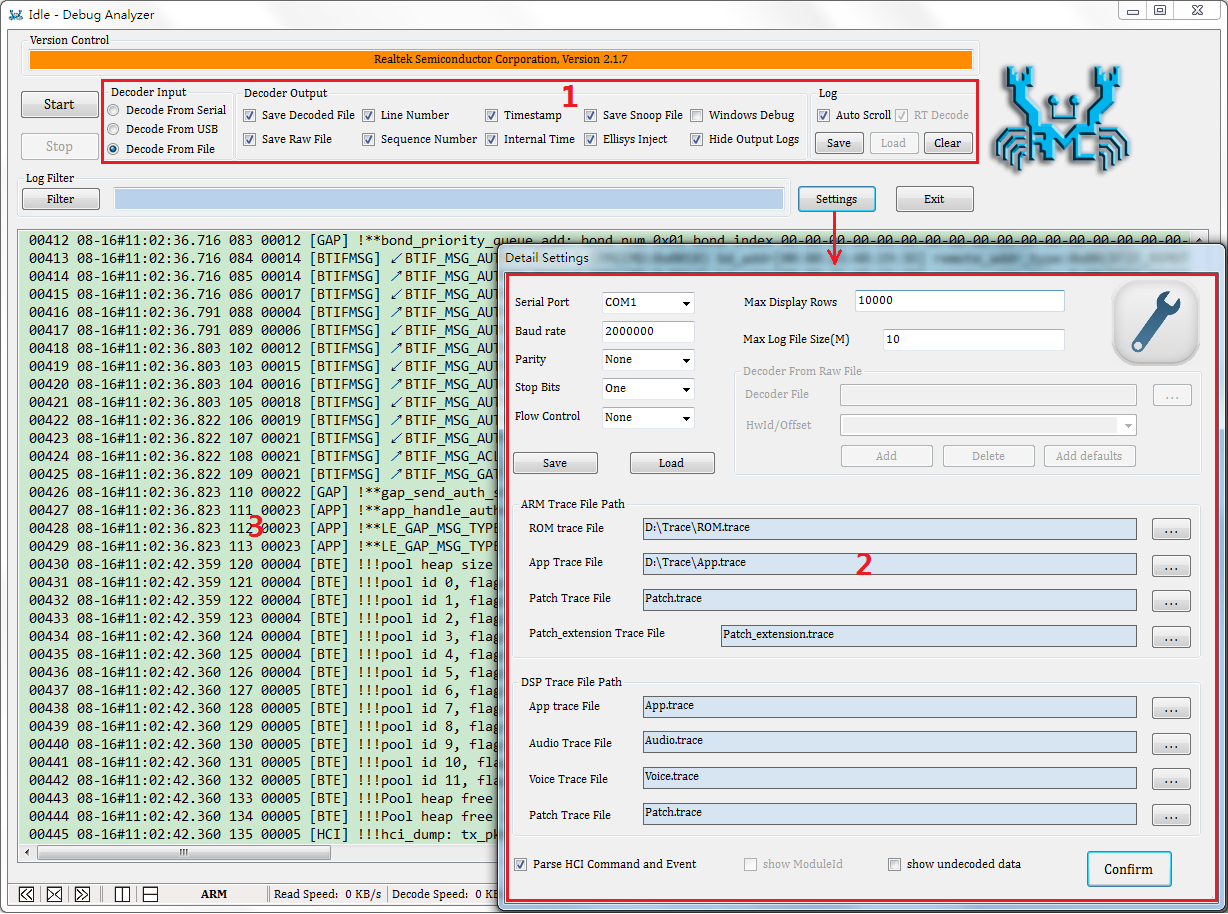


Figure 2‑1 Overview of DebugAnalyser

## Presentation of UI

Figure 2-1 shows the two essential dialogs: main dialog and “Detail Settings” dialog.

**Area 1** contains the input and output settings.

**Area 2** contains the detailed settings for log decoding.

**Area 3** is used for displaying decoded logs.

## Preparing

If real time log capture is needed, connect PC and SoC chip via serial port.

The raw data captured from SoC can be saved in a .bin file.

* If trace file is provided, the raw data can be decoded, and shown on UI. The decoded logs can be saved in .log/.cfa file.
* If trace file is not provided, you should check “RT Decode” on UI. The raw data will not be decoded but saved automatically. And you can decode it later if trace file is provided.

Figure 2-2 shows the **raw data file (.bin)**, **output log file (.log)** and **BT HCI log file (.cfa)**.

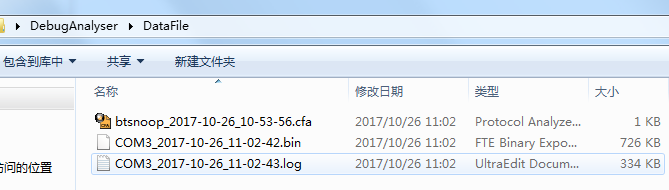


Figure 2‑2 Output Files

# Configurations of DebugAnalyser Tool

## Configuration of Main Dialog

1. Click to select one type of “Decoder Input”.

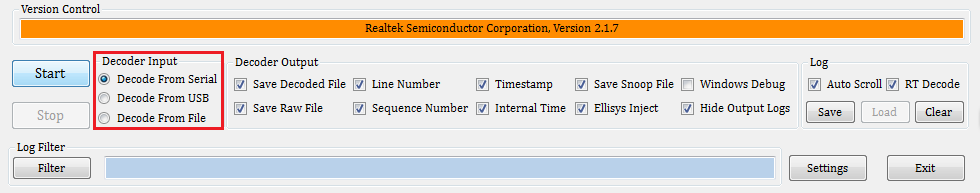


Figure 3‑1 Log Decoding Type

|  |  |  |
| --- | --- | --- |
| Option | Selection | Description |
| Decoder Input | Decode From Serial | If selected, the raw data to be decoded is captured from serial port. |
| Decode From USB | If selected, the raw data to be decoded is captured from USB. |
| Decode From File | If selected, the raw data to be decoded is captured from .bin file on local disk. |

Table 3‑1 Decoder Input selections

1. Decoder Output: Settings of decoder output, as shown in Figure 3-2.

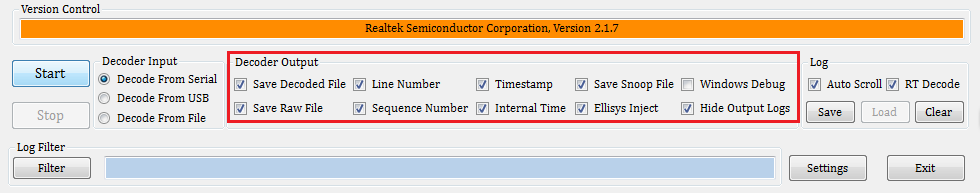


Figure ‑ Decoder Output Settings

|  |  |  |
| --- | --- | --- |
| Option | Selection | Description |
| Decoder Output | Save Decoded File | If checked, output logs will be saved in .log file.  If not checked, output logs will not be saved in .log file. |
| Save Raw File | If checked, the original data captured from bus is saved in .bin file. |
| Line Number | If checked, “Line Number” is included in the output logs. |
| Sequence Number | If checked, “Sequence Number” is included in the output logs. |
| Timestamp | If checked, “Timestamp” is included in the output logs.  Timestamp is the time at which log is decoded. |
| Internal Time | If checked, “Internal time” is included in the output logs.  Internal time is the timestamp provided by BT SoC chip. |
| Save Snoop File | If SoC chip provides HCI information to application, .cfa file is generated when this button checked. More details in [5.1](#_Open_DebugAnalyser_Tool). |
| Ellisys Inject | If checked, HCI data will be send to application “Ellisys”.  More details in [Chapter 5](#_How_to_use). |
| Windows Debug | If checked, display and save this tool’s debug information in a .log file.  If not checked, do not display and save the debug information. |
| Hide Output Logs | If checked, do not display the output logs on UI. |

Table 3‑2 Decoder Output Selections

1. Log

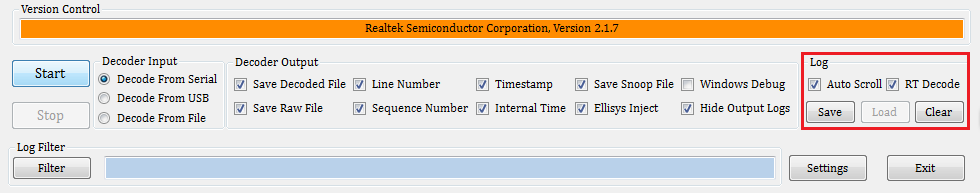


Figure 3‑3 Log Settings

* When “Auto Scroll” is checked, logs shown on UI will scroll automatically when decoding.
* When “RT Decode” is checked, raw data will not be decoded.
* Click “Clear” to clear logs shown on UI. Logs saved in files are not affected.
* Click “Save” to save logs currently shown on UI to a file.

## Configuration of Detail Settings Dialog

As shown in Figure 3-4, the “Detail Settings” dialog comprises the following functional areas which can be configured to definite the decoding process.

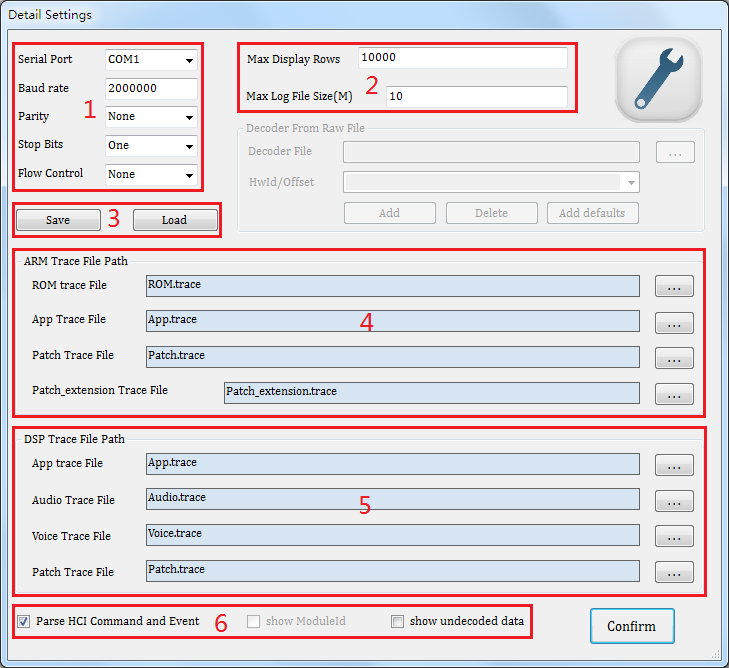


Figure 3‑4 Detail Settings

If “Decode From Serial” is selected, make sure to configure “serial port(area 1)”.

|  |  |  |
| --- | --- | --- |
| Area | Setting | Description |
| 1 | Serial Port | Set serial port number. |
| Baud rate | Set the baud rate of data transfer. |
| Parity | Set the mode of parity. |
| Stop Bits | Set stop bits. |
| Flow Control | Set the mode of flow control. |

Table 3‑3 Serial Port Settings

Configure area 2, 4, 5 and 6 as needed.

If there is no .trace file, the default value will be used for decoding.

|  |  |  |
| --- | --- | --- |
| Area | Setting | Description |
| 2 | Max Display Rows | Set the maximum rows of log display on UI.  If logs shown on UI exceed the max rows, empty the display area. |
| Max Log Size | Set the maximum size of log file.  If log file size exceed the max size, a new log file will be generated. |
| 3 | Save | Save the configuration parameters as a .cfg file. |
| Load | Load the configuration parameters. |
| 4 | ARM Trace File Path | Set ROM, App, Patch and Patch\_extension trace file path for decoding ARM log. |
| 5 | DSP Trace File Path | Set App, Audio, Voice and Patch trace file path for decoding DSP log. |
| 7 | Parse HCI Command and Event | If checked, decode HCI command and event of RTL8762A. |
| Show undecoded data | If checked, show the data not decoded. |

Table 3‑4 Detail Settings

Click “Confirm” to save above settings.

## Start and stop capturing and decoding

Click the “Start” button to start decoding.

If “From Serial” is selected, please make sure serial port settings are configured (refer to [3.2](#_Configuration_of_Detail)).

If “From File” is selected, a dialog will pops up to guide you configure the file name and position (Figure 3-5). Only if a raw data file (.bin) has been selected, the tool will start decoding.

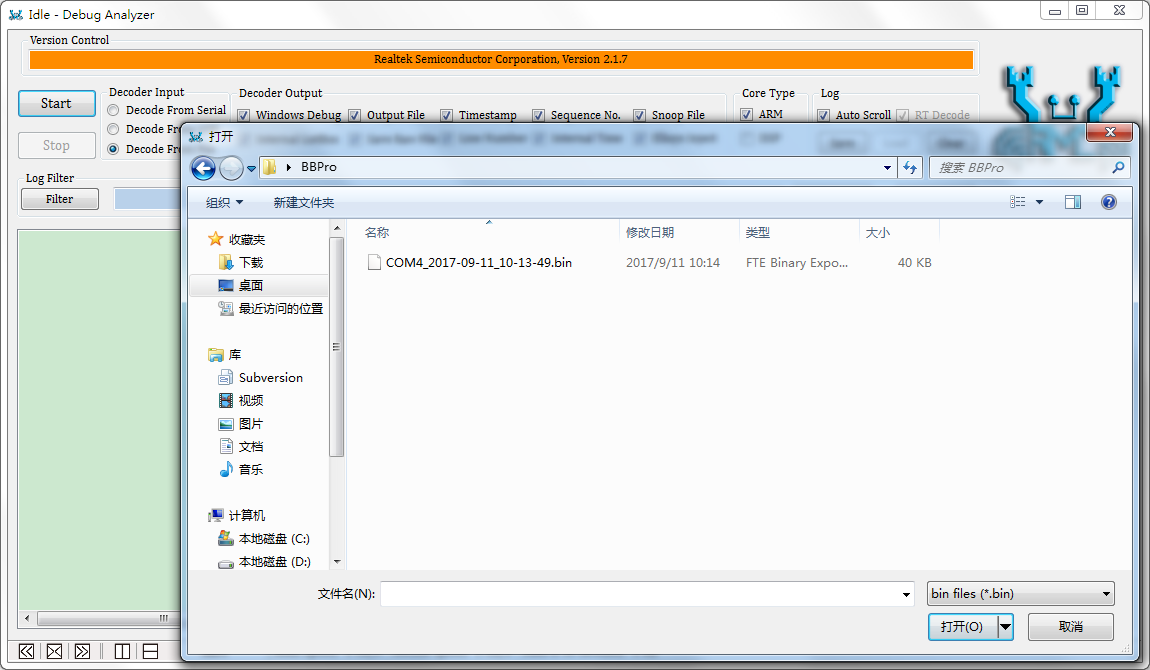


Figure 3‑5 Select Bin file

Click “Stop” to stop decoding.

# Log Operation and Display on UI

## Log Operation on UI

You can copy, search and filter the logs shown on UI.

1. Copy logs

As shown in Figure 4-1, select the logs to be copied. Then press “Ctrl + C” to copy them.

1. Show details of one log

Double click on a log, dialog box will pop up to show details of the log, which is shown in Figure 4-2.

1. Search and filter logs

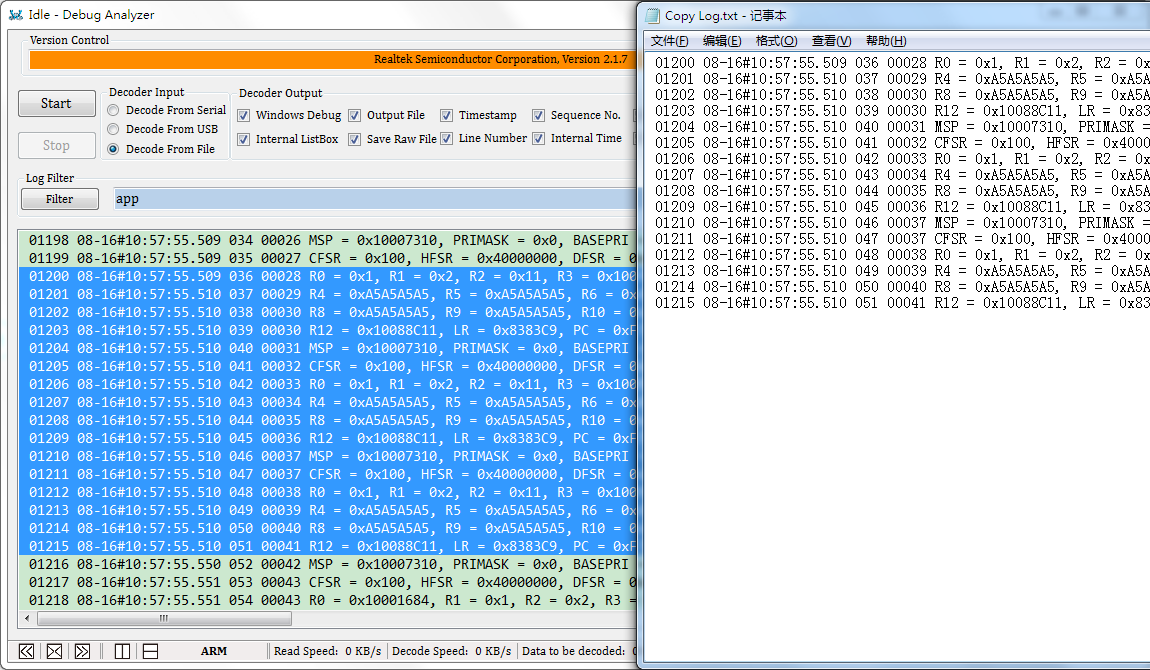
* Enter the keyword in the edit column of the filter.
* Click the button “Filter” or press key “Enter” on the keyboard, the filter reslut is shown in Figure 4-3.
* Double click one log of the Log Filter dialog, which can locate the log in the main dialog.
* Click button “save” to save.
* 

Figure 4‑1 Copy Logs

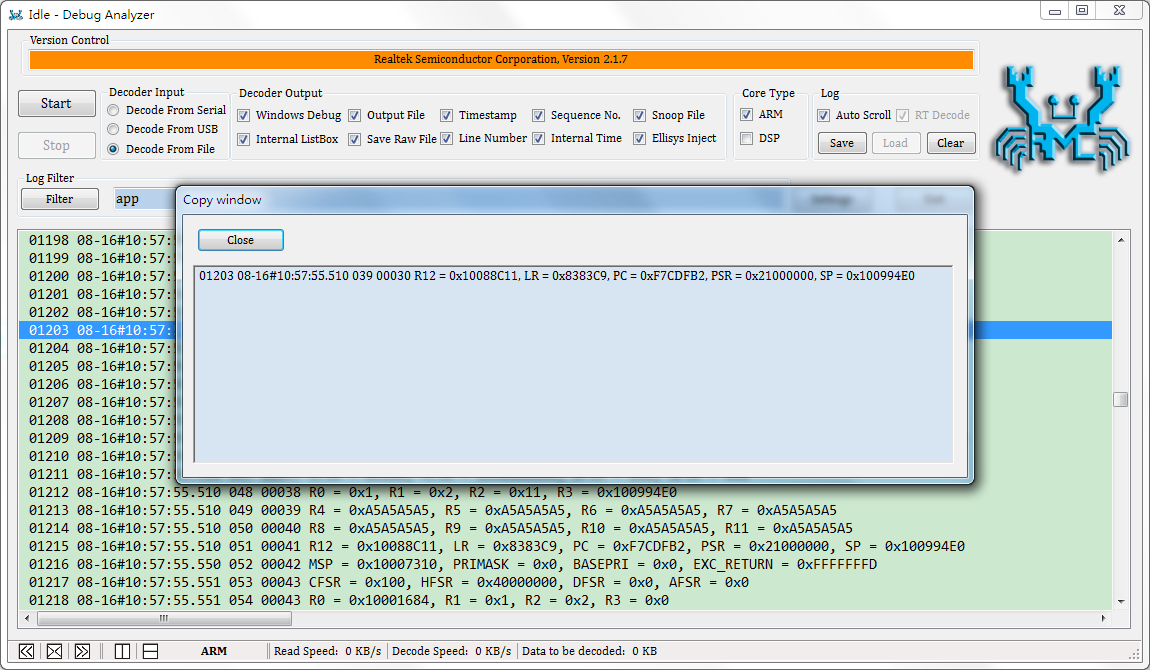
* 

Figure 4‑2 Show Log Details

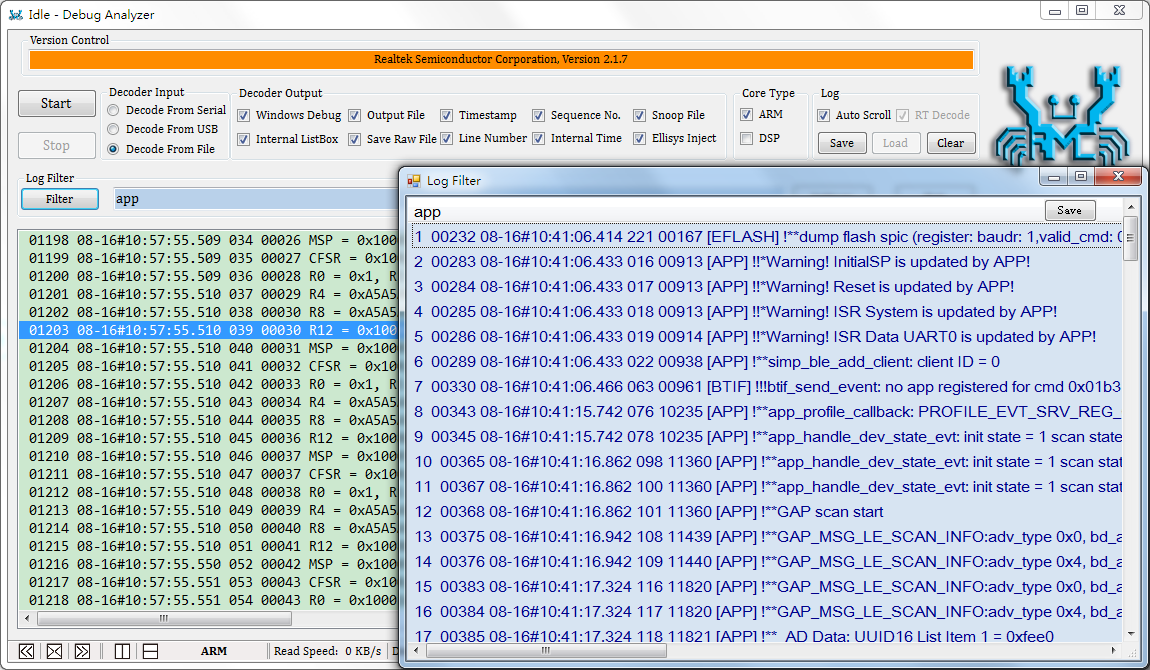


Figure 4‑3 Search and Filter Logs

## Log Display on UI

Three types of log display are shown as below.

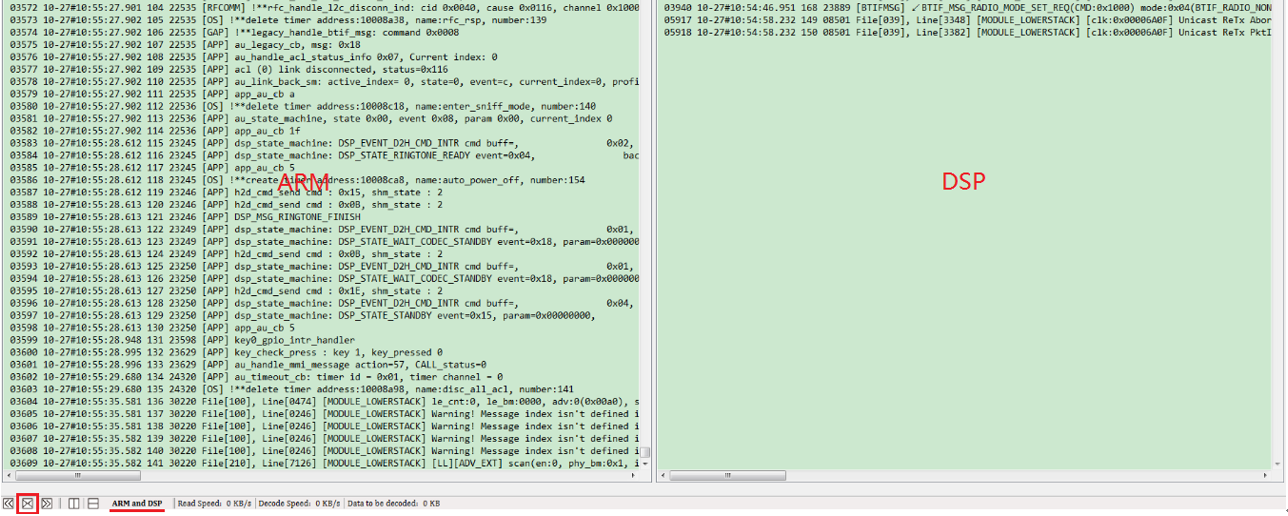


Figure 4‑4 Show ARM and DSP Logs

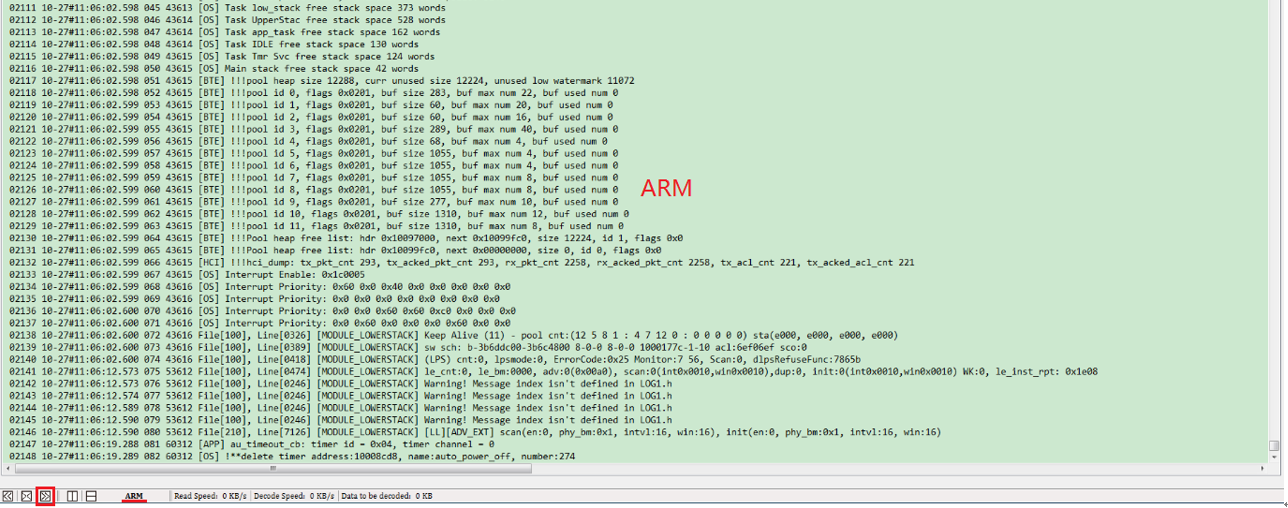


Figure 4‑5 Show ARM Logs Only

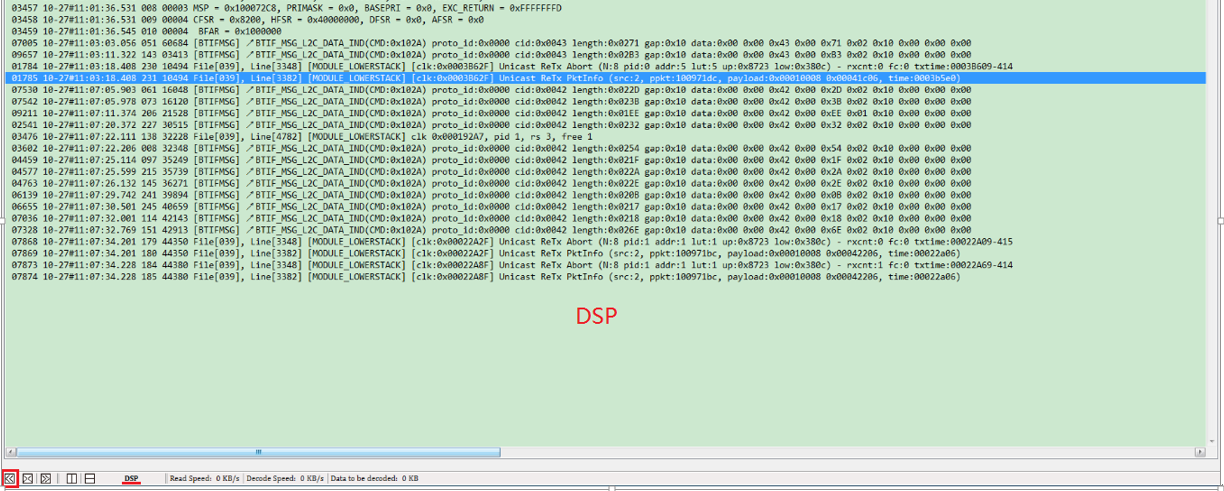


Figure 4‑6 Show DSP Logs Only

# How to use Ellisys view package of BTSnoop

The tool supports real-time view of BTSnoop log with Ellisys, which can view the real-time interactive packets of MCU HCI layer.

One should check “Snoop File” and “Ellisys Injection” in the “Decoder Output” area to use this function.

## Open DebugAnalyser Tool

Please ensure that the version of DebugAnalyser Tool is 2.0.0 or above. The process of generating BtSnoop Log file is as following:

* Configure the COM port in “Detail Settings” Dialog.
* Click “start” to start capturing the packets.

As shown in Figure 5-1, the BtSnoop log file is saved in the “DataFile” directory, and its naming format is “btsnoop\_year-month-day-hour-minute-second.cfa”. If the BtSnoop log file don’t exist, please check whether SWDTool is configured correctly.

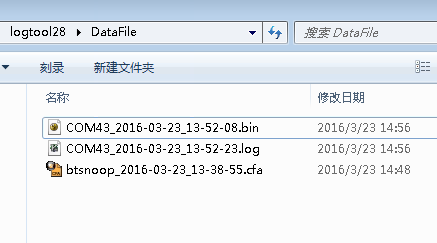


Figure 5‑1 DataFile directory

## The Process of Using Ellisys

1. Open Ellisys，as shown in Figure 5-2.

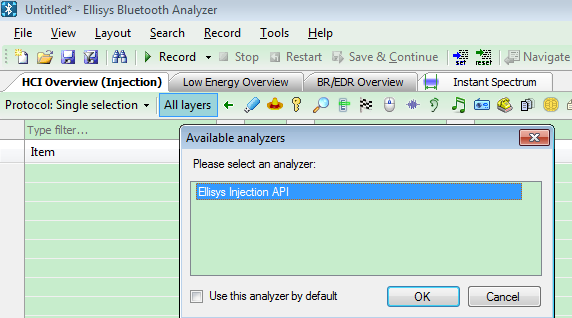


Figure 5‑2 Open Ellisys

1. As shown in Figure 5-3, one should click "Configure ..." to confirm that the” Classic Bluetooth” and “Bluetooth Low Energy” options are selected for the first time.

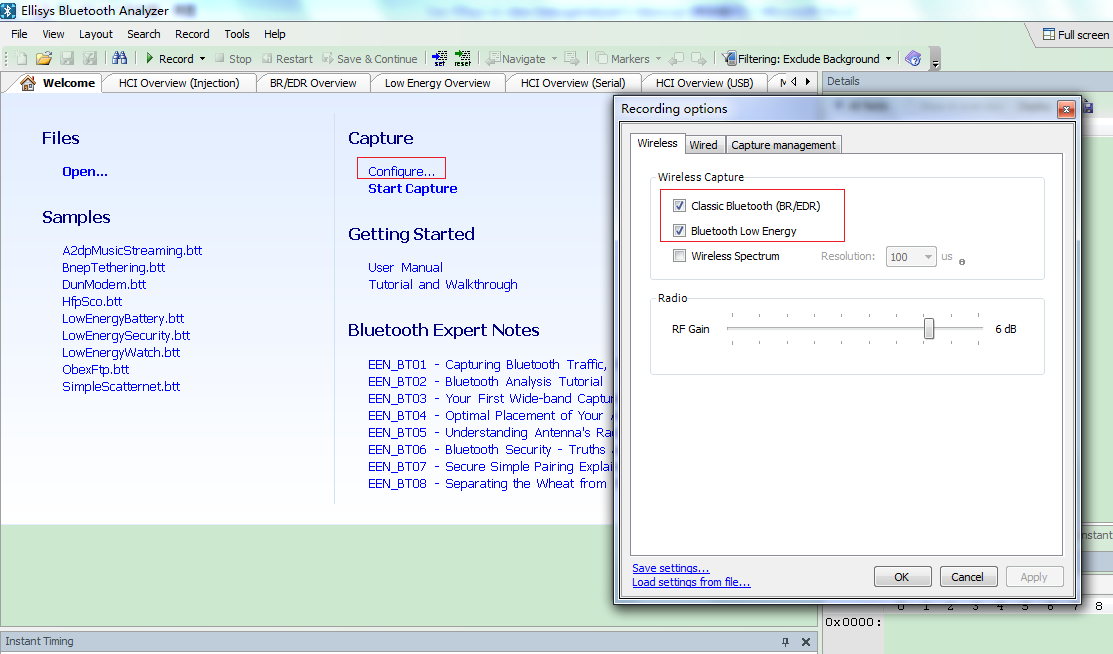


Figure 5‑3 Ellisys Configure

1. Click “Start Capture” to start capturing packets.

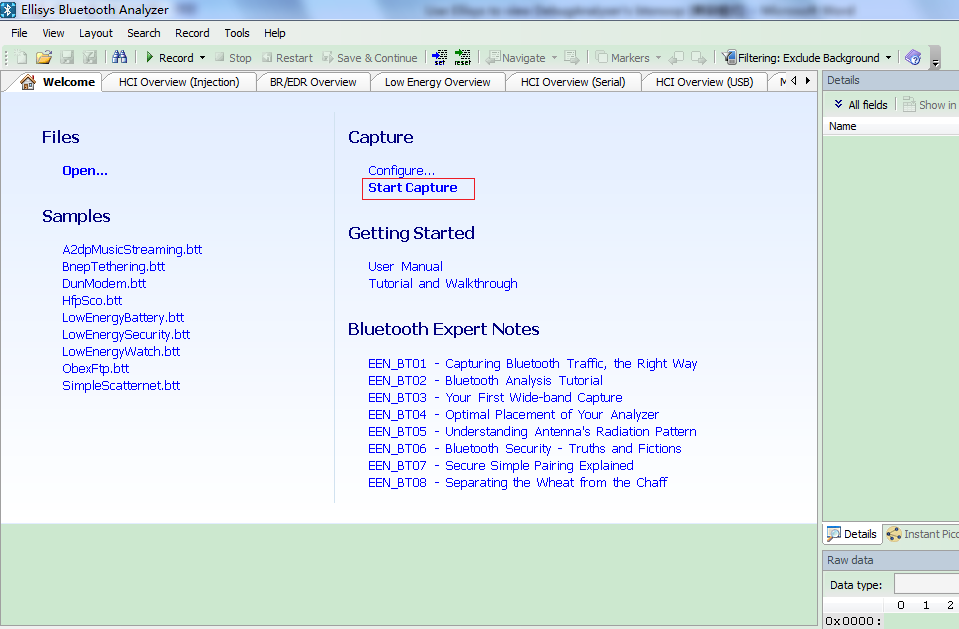


Figure 5‑4 Start Capture

1. Open HCI injection

Check View->Overviews->HCI Overview(Injection)，as shown in Figure 5-5.

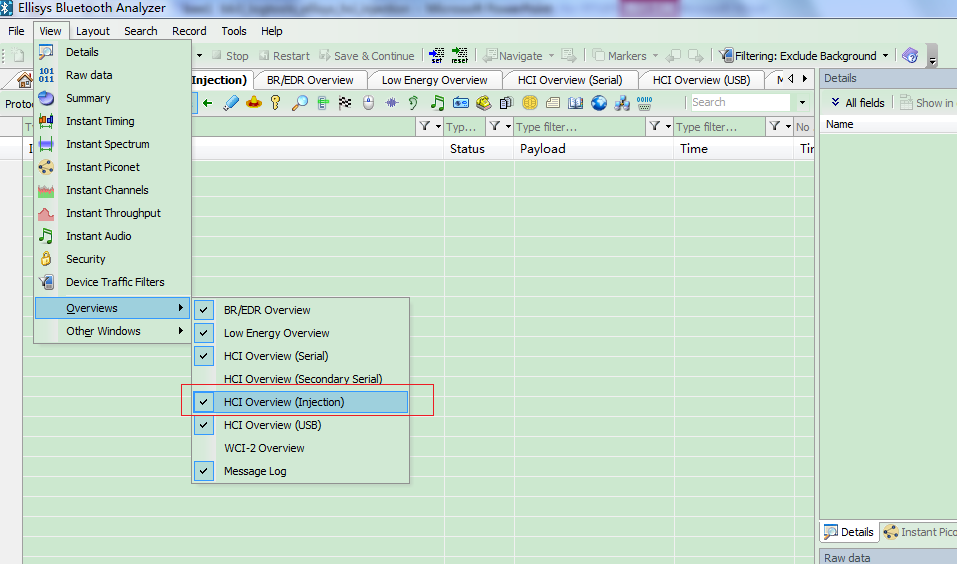


Figure 5‑5 Open HCI Overview（Injection）

The package of BTSnoop caught by the DebugAnalyser Tool is displayed in real time on the HCI Overview (Injection) dialog. Figure 5-6 is the contents after resetting Bee.

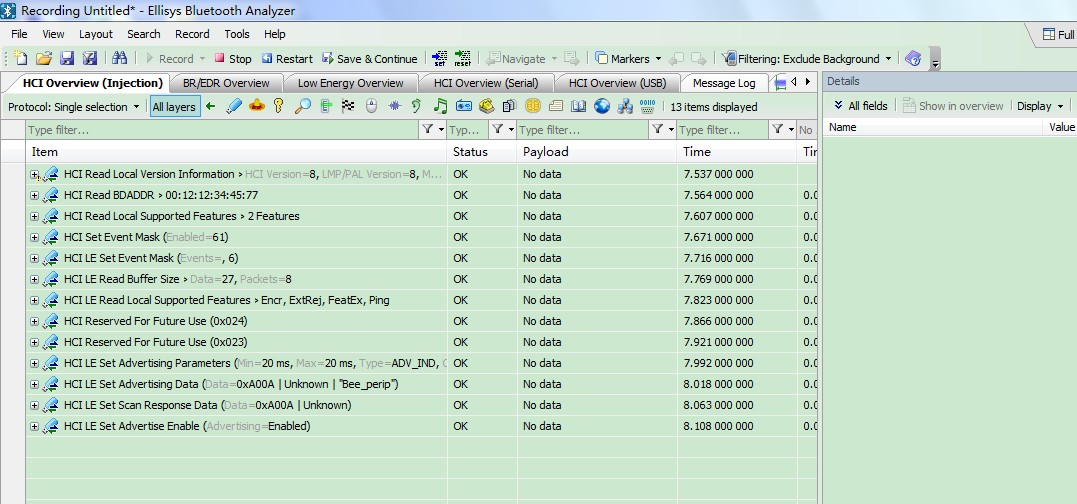


Figure 5‑6 HCI Overview（Injection）

## Notes for Using Ellisys to view BTSnoop

1. Ellisys does not affect the decoding state of DebugAnalyser. For example, DebugAnalyser is still in the decoding state after turning off or open Ellisys.
2. Ellisys air sniffer and HCI injection can be used at the same time.
3. Multiple DebugAnalyser tools run on one PC is not suggested, as HCI injection data may be mixed together.
4. The Ellisys function is only used to view HCI layer package.